THE ARMY AND Y2K: MANAGING THE COMPLEXITY OF TECHNOLOGICAL INNOVATION IN TACTICAL SYSTEMS

A MONOGRAPH
BY
Major Randall M. Bentz
Aviation

School of Advanced Military Studies
United States Army Command and General Staff College
Fort Leavenworth, Kansas

First Term AY 98-99

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	17 December 1998	Monograph

4. TITLE AND SUBTITLE
THE ARMY AND Y2K: MANAGING THE COMPLEXITY OF TECHNOLOGICAL INNOVATION IN TACTICAL SYSTEMS

5. FUNDING NUMBERS

6. AUTHOR(S)
BENTZ, RANDALL M., MAJ

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
School of Advanced Military Studies
Command and General Staff College
Fort Leavenworth, Kansas 66027

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Command and General Staff College
Fort Leavenworth, Kansas 66027

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
SEE ATTACHED

14. SUBJECT TERMS
STRATEGIC PLANNING; YEAR 2000 COMPUTER PROBLEM COMPLEXITY;

15. NUMBER OF PAGES
51

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Abstract

THE ARMY AND Y2K: MANAGING THE COMPLEXITY OF TECHNOLOGICAL INNOVATION IN TACTICAL SYSTEMS By Major Randall M. Bentz, United States Army, 40 pages

The Army operates in an environment of ever-increasing complexity. Traditional management models that have governed the actions of agencies and bureaucracies are becoming increasingly inadequate for dealing with the details and emergent realities of complex systems. The Army, like any other large government bureaucracy, must learn to adapt to the unfamiliar patterns of actions and products in a complex operating environment.

How the Army responds to a complex environment is evident in its planning process. This paper seeks to illuminate what the Army intends when it plans and what it means to plan in a complex environment. The Army's effort to remediate the Year 2000 computer problem (Y2K) is an example of both.

The Army's Action Plan for dealing with Y2K is the case study that this paper examines. It is a significant example of how the Army plans and provides insight into how the Army and the government understand planning in a complex environment. Through a series of performance controls, namely the Year 2000 database, the plan's timeline, and the Compliance Certification Checklist, the Action Plan is a method for controlling the process of Army Y2K remediation.

While the Action Plan itself follows a traditional organizational model that places emphasis on monitoring outcomes and adherence to a schedule of action, subordinate agencies governed by the plan follow a different organizational model; one that reflects the experience of complex systems management. The difference between the two approaches provides an insight into what is necessary to adequately manage complex actions in complex systems: a type of control that allows an organization time to learn while it accomplishes its mission.

SCHOOL OF ADVANCED MILITARY STUDIES

MONOGRAPH APPROVAL

Major Randall M. Bentz

Title of Monograph: *The Army and Y2K: Managing the Complexity of Technological Innovation in Tactical Systems*

Approved by:

William J. Gregor, Ph.D. — Monograph Director

LTC Robin P. Swan, MMAS — Director, School of Advanced Military Studies

Philip J. Brookes, Ph.D. — Director, Graduate Degree Program

Accepted this 16th Day of December 1998

Abstract

THE ARMY AND Y2K: MANAGING THE COMPLEXITY OF TECHNOLOGICAL INNOVATION IN TACTICAL SYSTEMS By Major Randall M. Bentz, United States Army, 40 pages

The Army operates in an environment of ever-increasing complexity. Traditional management models that have governed the actions of agencies and bureaucracies are becoming increasingly inadequate for dealing with the details and emergent realities of complex systems. The Army, like any other large government bureaucracy, must learn to adapt to the unfamiliar patterns of actions and products in a complex operating environment.

How the Army responds to a complex environment is evident in its planning process. This paper seeks to illuminate what the Army intends when it plans and what it means to plan in a complex environment. The Army's effort to remediate the Year 2000 computer problem (Y2K) is an example of both.

The Army's Action Plan for dealing with Y2K is the case study that this paper examines. It is a significant example of how the Army plans and provides insight into how the Army and the government understand planning in a complex environment. Through a series of performance controls, namely the Year 2000 database, the plan's timeline, and the Compliance Certification Checklist, the Action Plan is a method for controlling the process of Army Y2K remediation.

While the Action Plan itself follows a traditional organizational model that places emphasis on monitoring outcomes and adherence to a schedule of action, subordinate agencies governed by the plan follow a different organizational model; one that reflects the experience of complex systems management. The difference between the two approaches provides an insight into what is necessary to adequately manage complex actions in complex systems: a type of control that allows an organization time to learn while it accomplishes its mission.

Table of Contents

Introduction

A man received a notice from his credit card company stating that he owed $0.00. Assuming nothing was wrong, he ignored the notice. A month later he received another notice that he still owed $0.00 and must pay the balance immediately. He called the credit card company and was assured that it was only a "computer problem" and not to worry about it. The next month he received another notice that his balance of $0.00 was now two months overdue and if he didn't remit the balance, his account would be closed. Very annoyed, he wrote a check for $0.00 and mailed it in. The next week, when he tried to use his credit card for a purchase, he found it would not process through the store's point of sale device. When he called the credit card company, the customer service representative informed him that his check for $0.00 had caused the entire computer system to crash, fouling up everyone's account and causing his account to be cancelled. Several weeks later, he received a new credit card and account number. He went to buy his wife a birthday present. He had planned on getting her a computer. He bought her a typewriter instead.

This story, allegedly based on actual events, has become an "urban legend" among internet denizens that illustrates the complexity of our society and one individual's method for dealing with that complexity. Like that individual, Army personnel must deal with and operate in an increasingly complex world. Traditional rules that have guided the behavior of individuals, organizations, and systems are proving less useful

for addressing the complexities of modern military operations. It is more difficult today for the average Army leader to have a firm grasp on the intricate workings of the weapon and automation systems that he or she manages than was the case only 10 to 15 years ago. Automation in particular has significantly changed the way we do business, adding tremendous capabilities to our organizations while, paradoxically, making life both simpler and more complex at the same time. While we, as members of this Army, may at times be tempted like the man in the story to eliminate complexity from our environment, we must realize that it is only through mastery of complexity, not its elimination, that the Army can move forward and meet the challenges of the future.

Mastering complexity is not just the means for meeting the challenges of the future; it is itself the Army's present challenge. The complex nature of the Army's current systems frustrates the efforts of managers throughout the organization to predict and manage via traditional business practices. Past experiences with simple systems typically do not apply to current situations. The Army's method for overcoming ambiguity is to manage its environment through planning. When a tactical unit, an Army agency, or the entire Army prepares to meet a challenge, it does so by formulating a plan. However, the term 'plan' has multiple meanings; the precise meaning in a particular context matters. The goal of this paper is to illuminate what the Army intends when it plans and what it means to plan in a complex environment. The

Army's effort to remediate the Year 2000 computer problem is an example of both.

The Army's Action Plan for dealing with the Year 2000 computer problem (Y2K) is a significant example of how the Army plans and as such provides insight into how the Army and the government understands planning in a complex environment. The details of Y2K remediation are a direct reflection of the management model applied and reveal the extent to which action within the Army is governed by traditional models of governmental organization and action. The details of the Y2K remediation show that, while the Army has clearly recognized and articulated a need for the management of complexity, planners and executives have been and largely still are tied to simple, rules-based management models that fail to anticipate the unexpected consequences of complex actions in complex systems. The Army has structured its Y2K Action Plan (hereafter referred to as "the Action Plan") around the use of performance controls. The Action Plan thus, relies too much on centralized control for performance and does not take into account unexpected problems that occur when line managers implement modifications in complex systems.

The Army did not design its Y2K Action Plan by itself. The Action Plan is derived from the Department of Defense Y2K Management Plan that in turn came from other government agencies. It is an executive, top-down plan that calls for centralized planning and decentralized execution. It outlines duties and responsibilities concerning Y2K repair management

at each managerial level and mandates a timeline as well as specific reporting requirements. Is the Action Plan adequate? This cannot be known. That answer will be provided on January 1, 2000. However, it is possible to examine the plan, and determine how it addresses the complexity of technological innovation in a tactical automated system.

An examination of the overall organization of the Action Plan shows a rather traditional organizational model that places emphasis on monitoring outcomes and adherence to a schedule of actions. Within the plan, however, are the program activities of the responsible agencies. Their plans follow a different model; a model that reflects the experience of complex systems management. The difference between the two perspectives provides a tension that ultimately either advances or hinders remediation. In a larger perspective the Action Plan seeks to control performance while subordinate plans seek to affect performance. The end result is a picture of how planning is influenced by the desire to control performance.

Chapter 1: Background: The Road to Complexity

> *Nobody owns the problem. Bureaucracies aren't designed to solve new problems that cross over jurisdictional boundaries. Y2K does both and more. "There is no Y2K manual on the bookshelf, so it can't be mine" Everyone is fretting about his or her own mission critical systems but nobody in the government is claiming ownership of the nation's "citizen critical" systems.*[1]

> *Where are the editors? Anyone can write software but nobody edits it to conform to grammar. We've put together a global network in the last 20 to 30 years without any adult supervision.*[2]

The Year 2000 computer problem (Y2K) is a product, or manifestation, of a tradition of linear, rules-based management. Mainframe programmers in the nineteen fifties and sixties solved a short-term problem (limited computer memory and storage space) by formatting databases and executable code to accept and recognize only two-digit dates. What they did not take into account were the higher order consequences of their actions. Many programmers realized that the code and data would require changing before the end of the century, but they believed that either (1) the machines they were programming would be long-since replaced by century's end, (2) they themselves would be dead or retired, or (3) someone else would fix it.[3] The consequence of these considerations was adoption of two-digit dates as a defacto standard. The government, led by the Defense Department, formally designated 2 digit dates as the official standard in the mid-sixties.[4] Once established, the standard was largely forgotten simply because the system worked. The implications of this short-term solution would become apparent only as the

nation approached the century's end.

By the mid-nineties the scope and nature of the problem caused by the widespread use of 2 digit dates became apparent to the operators and managers of automated systems nation-wide. Among other government agencies, the Defense Department realized that it had become heavily dependent on automated systems and that Y2K could jeopardize the very foundations of national security. The Army, which had expended tremendous resources on developing its automation systems in anticipation of Force XXI, recognized Y2K as a serious threat to that development.[5] This case study focuses on one of the Army's new automated systems and how the Y2K remediation process affects it.

The Army has developed the Army Battle Command System (ABCS) to deal with the complexities of the tactical battlefield. ABCS is an automated network of devices and connections intended to increase the commander's situational awareness, speed up the decision making process, and integrate maneuver, logistics and fire support into a seamless whole. The system allows the commander to engage the enemy in the most efficient and effective way possible.[6]

Critical to the proper functioning of ABCS is the portion that delivers the commander's situational awareness, the All Source Analysis System (ASAS.) ASAS, a complex system of systems in its own right, receives intelligence data from multiple sources. It stores, collates and processes the data and presents them in various formats ready for analysis and

dissemination. ASAS consists of several different processors and software packages (All Source, Single Source, and Remote Work Stations) tied together by a telecommunications architecture (the Communications Control System,) serviced by Communications Electronics Command (CECOM) and managed by the Project Managers Office (PMO).[7] ASAS's designers fashioned the system in phases (blocks). Subsequent blocks replace previous blocks as time, technology, and resources allow, giving the tactical commander situational awareness that improves in an evolutionary manner.[8] ASAS was designed as a continuous system in which reliability, sophistication, effectiveness, and ease-of-use increase steadily throughout its lifecycle. Changes and improvements are supposed to take place in an orderly manner so as to minimize discontinuities and disruptions to units in the field.

The effects of Y2K

The year 2000 computer problem (Y2K) represents an unanticipated discontinuity that affects the orderly process of ASAS development and attacks the very basis of Army (as well as civilian and other governmental) automation. Processors that fail to recognize the century date change may calculate data incorrectly, corrupt databases with erroneous information, or simply shut down due to internal logic failures. Because ASAS is a system of processors tied together by a telecommunications network and connected to multiple information sources, it is very susceptible to receiving bad data from external inputs

and processing bad data internally.

CECOM and the ASAS PMO recognize this problem and are currently expending significant time and resources to fix it. Based on guidance from the Army's Y2K Action Plan, CECOM and the PMO are currently renovating and replacing hardware and software throughout the system. Testing takes place as each component is repaired with a full test of the entire system scheduled at Ft. Hood in March 1999.[9] ASAS, with as many of the associated Military Intelligence, other Army information sources, and Joint and National information assets as possible will participate in this comprehensive test, representing a complex amalgamation of systems that caps a long and complex remediation process.

Complexity

Large automation projects, whether new innovations or upgrades and repairs of existing systems (such as Y2K remediation) are classic examples of complex systems. Managing and keeping track of the repairs on vast numbers of lines of code, each with its own specific function poses a daunting task to any project manager. Large programs can be thought of as "discontinuous systems" that "will always harbor odd corners or a fatal response triggered by a one-in-a-million combination of input that eluded detection of both systematic and sample-based testing."[10] Large scale automation projects have historically experienced problems in the areas of timely completion and reliability. The complexity of these large

projects is in large part responsible for these problems. According to Ed Yourdon, an expert in mainframe programming with several programming textbooks to his credit, only sixteen percent of all large automation projects over the last thirty years have been completed on time and on budget. The remaining eighty-four percent have either been late, over budget, or cancelled.[11] Rather than being an indicator of the Information Technology (IT) industry's general competence (or lack thereof,) this large percentage of program failures demonstrates the unexpected results and unintended consequences of writing or renovating large amounts of code. Problems with complex systems, such as code remediation, "are inherently uncontrollable. Since prediction and control are impossible, traditional approaches to solve them simply won't work."[12] Large programming efforts are complex and difficult to manage, especially using traditional, machine-based management techniques.

The ASAS Y2K repair, a large and complex automation project, is just as likely to miss its deadline as any other automation project has in the past. The only difference in this case is the fixed and immovable deadline of 1 January 2000. If ASAS fails to meet this deadline, the ramifications will be serious. Commanders may be isolated from critical intelligence at all echelons and may fail to win the battlefield information war.[13] Full testing of the entire system is a long, complex, and laborious process, involving the participation of commanders, uniformed personnel, Army civilians, Army contractors, and independent civilian contractors.

The tests must exhaustively check all aspects of the system's operation to reduce the number of errors and bugs to a manageable level.

The sum of these characteristics makes the ASAS Y2K project a suitable case study for examining how the Army plans for complexity and how a subordinate agency implements that plan. The Army's plan for controlling the overall Y2K remediation process and the ASAS project manager's methodology for solving Y2K within the framework of the Army's plan are the subjects of the next two chapters.

Chapter 2: The Army Y2K Action Plan: The Simple Solution

> *Planning is a means of reducing external complexity to manageable forms.*[14]

> *Planning was used as a tool by top management to try and regain control of its organization.*[15]

A centrally developed plan is a mechanism to control an organization. Traditionally, planning represents a formalized process. Planners and executives develop a plan that identifies a problem, establishes organizational goals, assigns responsibility, and allocates resources. Henry Mintzberg, a management educator and a former president of the Strategic Management Society, sees in this traditional understanding of planning a methodology rooted in analysis. "Decomposition of the process of strategy making into a series of articulated steps, each to be carried out as specified in sequence, will produce integrated strategies." In other words, "analysis will produce synthesis."[16] By merely breaking the problem into its component parts, planners can understand the problem through its parts and thus control the entire process.

A plan developed under a linear, mechanical model envisions the organization's actions throughout the life of the planned event. Actions occur in a logical, sequential order. The organization's success or failure becomes apparent based on concrete performance control measures at certain junctures in the event. If, based on a certain action at a certain time, line managers are supposed to produce a certain result and fail to do

so, the assumption that the planners and executives normally make is that the line managers were not following the plan. In an industrial assembly line, this might well be the case. In a complex environment, however, there are other factors that alter the plan and may make expected performance measures unrealistic. A formalized plan operating in an adaptive environment can breed inflexibility that will not allow change and can cause central executives and planners to quickly lose control of the entire event. The Y2K Action Plan contains some elements of formalization that are problematic.

The Army Y2K Action Plan is a comprehensive document, produced by the Army Y2K Project Office under the guidance of the Director of Information Systems for Command, Control, Communications, and Computers (DISC4.) The DISC4 has overall responsibility for the Army Y2K effort. The Action Plan outlines the Army Y2K management strategy, provides guidance, defines roles and responsibilities, defines reporting requirements, and establishes a framework to ensure that no mission critical systems fail due to Y2K. It applies to all systems in the Army supported by information technology, their technical environment, and their communication devices.[17]

The term "Army Action Plan" is really a misnomer because it isn't so much an action plan as it is a series of performance controls. According to Mintzberg, elements of performance control "are routine in nature, logically carried out on a regular basis, quantitative in approach

and largely the concern of the accounting people."[18] Performance controls influence the organization indirectly by establishing budgetary limits or other parameters. In the case of the Y2K Action Plan, the performance controls are (1) the Y2K database, (2) the timeline, and (3) the compliance checklist. A true action plan, on the other hand, is a "before-the-fact specification of behavior"[19] which is less quantitative than a budget and more the purview of the line managers, supported by the planners. It is a strategy that prescribes the execution of actions, much like the concept of operations in a military operations order. The Y2K Action Plan gives no such clear guidance, but rather, establishes an end state and imposes boundaries.

Realistically, the Action Plan cannot provide clear guidance for all cases, because it is written to provide a framework. The Action Plan establishes a series of requirements leading to an end-state (Y2K compliance), and therefore is not a true plan for action. In this sense, it should not be called an action plan at all. The DISC4 called it what it is, in all likelihood, because of the positive and active connotations associated with the term "Action Plan."

The Army Y2K Database (Y2KDB)

The purview of the Action Plan is tremendously broad. It covers every mission critical system in the Army that Y2K may affect. The Action Plan defines mission critical systems as those systems that support intelligence activities, cryptologic activities, command and control of

military forces, weapon or weapon systems, and other systems critical to direct fulfillment of military or intelligence missions[20] (ASAS qualifies as a mission critical system by all of these criteria.) It is unrealistic to expect the DISC4 to be conversant, let alone intimately familiar, with all of the Army's mission critical systems governed by this plan. Instead, the DISC4 has instituted a performance control measure known as the Army Y2K Database (Y2KDB) which receives data from every Project Manager on the status of his or her Y2K remediation program.[21] This flow of hard data up from the various organizations and agencies into the Y2KDB allows the DISC4 to remain abreast of the latest general information concerning Army Y2K progress. There are two problems with this. First, this process may create the mistaken notion that by knowing what's currently happening in the overall Y2K remediation process, the Army can control the process and make decisions from a central location. This is a planning fallacy that Mintzberg refers to as "the assumption of quantification." Strategy is driven by hard data, comprising quantitative aggregates of detailed "facts" about the organization and its environment.[22]

The assumption of quantification necessitates the regular flow of hard data up the hierarchy for the executives and planners to generalize and make into strategy. In order to not overly burden Army PM's with undue reporting requirements, some of the information input into the Y2KDB has been reduced to "one liner"[23] entries. While providing easy

reference to both PM and DISC4, this information is by its nature "limited in scope and does not encompass the totality of the environment."[24] Because abbreviated data, even if precise, cannot convey the whole picture of an agency's remediation efforts it can lead to an inaccurate understanding of the situation. As Mintzberg relates, "quantitative measures are a crude surrogate of reality."[25] This implies that any decision made based on Y2KDB data may not be the best for a particular system because the decision would not be taking all of the pertinent environmental data into account. Conversely, if the Army doesn't intend to make decisions based on Y2KDB data, then the Y2KDB is merely a control device for information purposes only. As such, it is a drain on the individual PMO's time resources and adds little benefit to the overall remediation process. The only real way that the Y2KDB can be useful is as a mechanism for raising a flag when an individual project is behind schedule, allowing the DISC4 to contact the PMO and discuss that particular situation in order to get the project back on track.

The other problem with the Y2KDB is while the Action Plan calls for close horizontal coordination involving Memoranda of Agreement between various agencies, especially those with interfacing systems[26], there is no entry field in the Y2KDB showing how many interfaces a system may have, what the nature of the MOA's are, and whether or not the MOA's between interfacing agencies fit within the overall guidelines of DISC4.[27] This could lead to an unclear picture of a system's compliance within its

external system of interfaces.

Timeline

The Action Plan has mandated a schedule for fixing the Year 2000 problem in Army automated devices. The timeline is spelled out in annex B of the Action Plan. The total remediation process consists of five phases. The first phase concerns awareness, education, and initial organization and planning. The completion deadline for this phase was 31 December 1996. The second phase is assessment. In this phase, the scope of the impact is identified and device and system level analyses take place. The completion date for this phase for all Army systems was 31 March 1997. The third phase, renovation, is where all required system and device fixes take place. The scheduled completion date for this phase was 30 September 1998. Phase four, validation, serves to confirm all system and device fixes are Y2K compliant through assorted testing and certification processes. The required completion date for this phase is 31 December 1998. In the final phase, implementation, all systems are fielded and fully operational after being certified. This phase will also be completed on 31 December 1998.[28] Graphically, the timeline looks like the following:

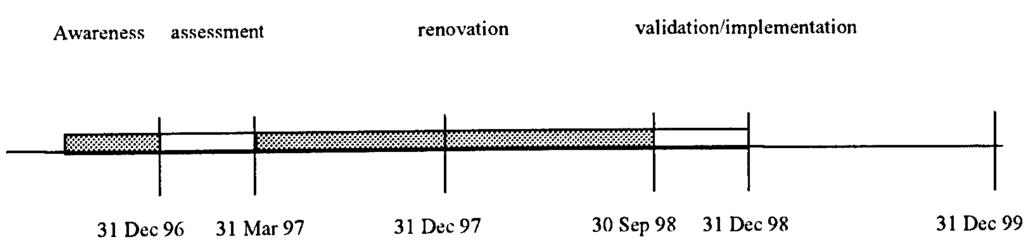

figure 1, Army Y2K repair timeline

Looking at the timeline in figure 1, it is apparent that the bulk of the available time has been dedicated to renovation. Of the two-year period between 31 December 1996 and 31 December 1998, Eighteen months, or seventy-five percent of the available time is for renovation. Awareness has been allotted 12.5 percent of the available time, while testing and implementation receives the remaining 12.5 percent. Understanding that even though a certain amount of testing will take place during the renovation phase, the individual Project Manager cannot run a full test of the entire system until the entire system has been renovated.

There are two other timelines that provide a basis of comparison for the Army's schedule. The first timeline is based on the findings of the California state government's White Paper[29], published 16 October 1996. The Gartner Group, a technology think tank, first suggested this timeline as a model for system automation project success. The California State government adopted it and, over the past two years, this timeline has become the unofficial standard for all government and corporate Y2K projects nationwide. It was based on the experience of various government and industry IT departments in dealing with large automation renovation projects. It breaks the Y2K remediation process into an eight-phase operation, lasting approximately thirty-nine months for a "large" project (ten million or more lines of code).

The first phase, awareness, takes one month and represents one percent of a project's cost. The second phase, inventory, also takes one

month and represents another one percent of the cost. Assessment is the third phase, lasting two months and consuming five percent of the project's budget. The next two phases deal with renovation. Solution design and planning takes four months and fifteen percent of the cost while development and modification lasts up to eight months and represents twenty percent of the cost. Testing, the largest single portion of the remediation process, takes up to eight months and forty percent of the project's resources. Implementation represents five months and ten percent and, finally, monitoring runs up to ten months and eight percent of total cost. Based on the assumption that organizations with large projects must begin work in late 1996 in order to be completed by December 1999, a timeline of a generic Y2K project under these guidelines looks like this:

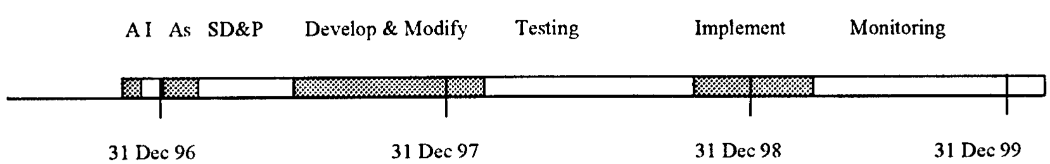

figure 2, California White Paper Timeline

Compared to figure 1, figure 2 shows a greater requirement for testing, forty percent of available resources and about twenty two percent of available time. The time dedicated to testing in figure 1, as outlined in the Army's Action Plan, is inadequate in light of what passes for a common standard in government and industry.

There is no apparent reason why the Army's timeline differs so greatly from that of the California state government or industry. The current revision of the Army Action Plan, dated June, 1998, should have

taken the additional requirement for testing into account and allotted more time for it. The plan, however, does not. This may indicate that the timeline was never realistic in the first place, or that the plan's authors didn't consider strict adherence to the timeline as important as overall project completion. It may even be the case that, by the summer of 1998, the Army's Y2K planners realized that Y2K testing is much more complex and time-consuming than originally thought, making even the White Paper timeline model insufficient. A third timeline model that supports this proposition appears, based on actual reports from the field.

Recently, based on their experience with actual Y2K project tests, several industry programmers have come forward and related that the California White Paper standards do not allocate enough time for testing. Their experience has shown that full up system testing takes considerably longer than anticipated, mainly due to unforeseen errors in code brought about by the Y2K repairs. In several messages posted on Ed Yourdon's web site[30], these programmers relate that "the system never tested out fine right away"[31] and that the accepted model is inadequate:

> ...there is a trap, into which, we have fallen. We presently divide our progress board into the stages of Inventory, Assessment, Remediation, and Testing. But the first two don't count for much. And Remediation results can be misleading because one won't know if he has remediated everything until the testing starts. We are using a four stage paradigm that does not tell us if we are making progress....Here is a cut at a more realistic paradigm that keeps us honest:
>
> I. Inventory, assessment, and remediation

II. Sub-system testing. Fix what you find. Retest

III. Regression testing. (Once A, B, and C work then get A and B working together and then A and B and C working) Fix what you find. Retest.

IV. "Full up" testing without tele. Tele testing. Then both together. Fix what you find and then retest.[32]

As this new paradigm implies, testing requires considerably more time than either the Army Action Plan or the California White Paper indicate. The issue for planners is to determine exactly how much time testing actually takes. Experience from industry gives some clues.

Several civilian organizations have come forward with new estimates on time required for testing. Data Dimensions, a software testing firm, states that "Year 2000 testing can consume seventy percent of a project's time and money."[33] The New Zealand Bankers' Association, dealing with their own Y2K remediation efforts, relates a similar experience:

> Gartner Group originally predicted that Year 2000 work would be 40 per cent repair, 45 per cent testing, and 15 per cent project management. For us it has been 10 to 20 per cent repair, 80 per cent testing, and the project management component is very small.[34]

Using this information, one can construct a third timeline as follows:

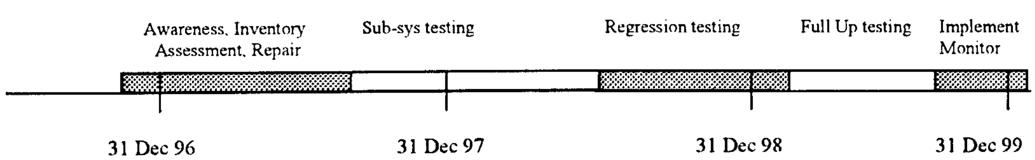

figure 3, Increased Testing Timeline

There are several conclusions that one can draw from this third timeline. First and most obvious, there is not enough time remaining until

December 1999 to fully test large systems. Even allowing that sub-system testing occurs during remediation, all repairs would have to be completed by mid-1997 to allow for enough testing before the end of 1999. ASAS did not begin its repairs until January 1998 and so it is already beyond this schedule. The "full up" test of ASAS is scheduled for June 1999.[35] CECOM and PMO ASAS will have only 6 months to fix any problems that surface during this series of tests before fully implementing the system Army-wide.

Why is the Army Action Plan so out of touch with the reality of testing? The answer likely lies in the nature of Y2K itself. Y2K represents an unprecedented challenge in code repair and testing. When programmers repair non-compliant code, several things must happen. The code itself must be made compliant, the repaired code must be compatible with the existing software and data within the system, and it must be able to function with the software and data from all of its interfaces with external systems. This is the essence of a complexity. The combined effects of a system are greater than the sum of its individual parts. If the testing process ongoing in civilian industry is any indication, the remediation process itself is responsible for many of the problems encountered in the validation phase. Ben Levy, the vice-president of an Israeli information technology group, has stated "In several cases we found date issues were either missed, not converted or converted wrongly. The problem is that one mistake in one program can

cause a major problem to a business."[36] David Marshall, managing director of Greenwich Mean Time, a British IT diagnostics company specializing in Y2K solutions, confirmed similar findings. "Too many companies have attempted to tackle the millennium bomb through a piecemeal approach to their systems."[37] By approaching each mission critical system as a stand-alone and not focusing on the interactions between systems, the Action Plan falls short in its ability to anticipate the consequences of code repair in an integrated environment.

In all fairness, the Action Plan does emphasize the need for managers of linked systems to establish Memoranda of Agreement between their agencies[38] that delineate message format requirements and compatibility issues, but, other than establishing a standard message format, it leaves the details up to the individual systems owners. Decentralizing authority in this manner is not necessarily a bad thing, but in a case like this, where the government, DoD, and the Army are trying to establish a standard format, decentralization without adequate control can lead to inconsistencies throughout the entire system. It also serves to negate any control over the interfacing process that DISC4 may have implied in the Action Plan. This would leave DISC4 with only the illusion of control over the process.

The Y2K Compliance Certification Checklist

The last significant performance control measure in the Action Plan is the compliance certification checklist, a comprehensive list of functional

tests and requirements that each mission-critical system must undergo in order to be certified "Y2K compliant."[39] Project managers must perform what the Action Plan refers to as a "positive test" on their respective systems. Any system controlled by the Action Plan cannot be considered compliant until it functions correctly by itself and "until all interfaces properly receive date related data and a completed Compliance Certification Checklist has been signed by the appropriate authorities."[40] An "appropriate authority" is defined as either a General Officer or a Senior Executive Service civilian associated with the project.

Given the level of individual responsibility and authority required to certify a system, it would appear that the compliance certification checklist is an effective method for controlling the Y2K remediation process. Independent audits by the Defense Department Inspector General, however, indicate that the checklist is not as effective a control as DISC4 probably envisioned. During an audit of systems throughout the Defense Department in June, 1998, the IG reported that

> DoD components are not complying with Y2K certification criteria before reporting systems as compliant. Of the 430 systems reported by DoD as Y2K compliant in November 1997, we estimate that DoD Components certified only 109 systems (25.3 percent). As a result, DoD management reported as Y2K compliant systems that have not been certified. More importantly, mission-critical DoD information technology systems may unexpectedly fail because they were classified as compliant without adequate basis. The results are based on a randomly selected sample of 87 systems that DoD had reported as compliant. A signed Y2K compliance checklist was requested for each of the systems selected.[41]

This report indicates that project managers, department, and agency heads either reported compliance on systems that did not have a completed checklist or that they signed checklists on systems that had not been adequately tested.

Incorrect reporting on system compliance may indicate a simple lack of integrity at senior management level; more likely, however, it indicates that the nature of Army mission-critical systems is so complex that the managers and even the programmers did not fully know the extent of noncompliance within their computer systems. Whether these reporting discrepancies occurred due to deceit, subterfuge, or honest mistakes is unimportant. The fact that these checklists proved unreliable in insuring system compliance degrades their value as performance control measures.

If a centralized plan is to have any success in controlling the behaviors of an organization, its control measures must be effective. Monitoring and enforcing the organization's response to the control measures is vital in ensuring that unit and individual behavior conforms to the plan. If the performance controls are inadequate, however, then the central authority's only option is to overlay more controls on the system. In a complex environment this approach can cause further unintended consequences. In the case of the compliance checklists, DISC4's response to unreliable certification papers may be to institute more detailed and rigorous reporting and auditing requirements on the

MACOMS, program executive officers, and project managers. Increased reporting and inspection requirements, however, are a drain on a project manager's already precious time resources. The more time the manager spends reporting on project progress, the less time is available for actual project management. This lessening of available supervisory time may result in increased repair or testing times, or other related problems.

The problems associated with overlaying extra controls in a plan represent a dilemma of ever-decreasing returns on control in a complex system. Army-level planners and senior leadership must face the choice between minimizing control, which may result in actions that deviate from the plan's intent, and maximizing control which, in a complex environment, will result in unintended consequences that cascade throughout the system.

Chapter 3: ASAS Y2K repair: Complexity in Action

> *What is interesting about the complexity of physical systems lies in their details – how they are put together.*[42]

When discussing a complex action (such as remediation) in a complex system (such as ASAS) it is helpful to have some sort of model with which to make comparisons and evaluations. Literature on complexity and the systems approach, however, gives little in the way of concrete models or examples. One exception is the book by Jason Kelly, editor of Wired magazine. In his book, Out of Control, he proposes a simple model for managing complexity of any type. It consists of starting small and simple and growing systematically larger and more complex. The six steps include:

1. Do simple things first
2. Learn to do them flawlessly
3. Add new layers of activity over the results of the simple tasks
4. Don't change the simple things
5. Make the new layer work as flawlessly as the simple
6. Repeat, Ad infinitum[43]

This implies, as was demonstrated in the last chapter, that comprehensive testing at the user level is an absolute necessity to ensure project success. It also implies that as each new layer of complexity is added, unexpected results appear which require new solutions or more time to test and meld into the whole project.

A timeline can serve as a starting point for a remediation plan, but

the planners and the operators must realize that the unexpected will occur. This manifestation of the unexpected requires planners to factor in additional time for testing. Rather than occurring in a sequential, linear fashion after repair, testing must take place concurrently with code repair in order to fully validate the repaired code, work out the "bugs", and allow the operators to proceed to the next level of complexity.

This model is quite different from that of the Y2K Action Plan. It does, however, correspond surprisingly well to the actual repair process going on in the subordinate agencies, particularly ASAS. The ASAS repair project involves a complex system being remediated according to the rules of complexity.

ASAS Block I is an interconnected system of systems consisting of processors, operating systems, vendor-supplied applications, and connections. It is a relational database that receives and stores data from external sources. It is also a powerful suite of software that processes the data, converting them into usable information that analysts can use to satisfy the commander's intelligence requirements. ASAS Block I represents a better remediation case study than Block II for a number of reasons. First, Block I represents a system upgrade from a noncompliant system to a compliant one. Block II is being designed as a compliant system, or at least modified during early development. Second, PMO ASAS can delay Block II fielding until it is completely compliant (Block I is already fielded, providing intelligence fusion for all tactical units.) Block I,

however, must be Y2K compliant by the deadline of 1 January 2000, because there is no backup system. Finally, ASAS Block I, which will be in use throughout the century change, consists of several more components than Block II, adding to the complexity of its Y2K repairs.

The CECOM programmers working on ASAS Block I are taking the complexity of their system into account as they go through the process of remediation. As each team finishes a module of code, they subject it to standard Y2K compliance checks. Once these modules, which are called "code drops," test out satisfactorily, then the programmers tie the modules into their respective programs, retesting the module against other modules to insure compatibility and coherence within the greater system. Any errors, unexpected results, or other "bugs" cause the programming team to examine, debug, and modify the code until it performs correctly. The end result of these actions is a system of code in a program that performs on a specified platform to a specified standard.

When each of the separate platforms (Single Source, All Source, RWS,) function properly, they are then tested singly against the CCS to validate the component's ability to transmit data outward. Finally, all of the components are tested against each other in an intra-ASAS interoperability test to insure that the new software code does not corrupt or change any of the data passed between the boxes. After the system's logic and code are proven internally consistent, the programming teams will test the system against its external interfaces.[44] Through the use of

bridging and windowing techniques, the external information sources will be able to transmit data to ASAS which in turn should be able to read the data correctly and process it into usable intelligence information.

The ASAS Block I remediation methodology corresponds closely to that of Kelly's and the programmer in the last chapter who proposed the third timeline. It also differs significantly from the linear model and timeline proposed by the Action Plan. One possible reason for this is the fact that between the Army Action Plan and the ASAS PM there are several layers of intermediate supervision. CECOM and the Army Materiel Command (AMC) are the higher headquarters of the ASAS CECOM project manager and each has produced its own Y2K guidance. As with subordinate commands in a military organization, the plan becomes more refined and definite as it progresses downward. Another reason may be the fact that the repair teams actually doing the work on ASAS are experienced IT professionals who have worked in complex systems involving computer code creation and repair. They know through experience that code writing produces bugs and unexpected events in a complex system and realize that testing is the largest portion of their work. They must also deal with factors that the Action Plan does not clearly articulate. One such factor is close coordination with software vendors.

While the Action Plan mentions vendors, it treats them as almost a separate side issue that has little relevance to the plan's overall success. The Action Plan calls for all contractual language with vendors to include

Year 2000 compliance requirements[45], but doesn't consider how closely the vendors are tied to the actual remediation work. Vendors of the Sun Solaris Operating System, which the Single Source and All Source components rely on, for example, originally promised CECOM a compliant O/S in version 2.0. Seven versions later, they finally produced a compliant operating system that the CECOM programmers could use in the ASAS components.[46] The problem with this was that CECOM had to install and test the first version, find that it was non-compliant, and return it to the vendor for rewrite. The CECOM programmers then tested the next version, returned it, and repeated the process until, after seven attempts, they finally had a workable system. This of course caused a delay in the schedule and necessitated close and continual coordination between the ASAS programmers and the civilian vendor. Considering that the project manager had to work with vendors of not only the operating system, but also with several other vendors of commercial application software packages, the complexity of remaining responsible to a timeline while dealing with organizations outside of his control was significant.

A project manager's time is consumed in large part by schedule management and reporting requirements. In addition to the reporting requirements specified in the Action Plan, each level of command (AMC, CECOM) has its own reporting requirements and other control measures. Also, because ASAS is a critical system used Army-wide, every Unified Command repeatedly requests information on the remediation's

progress.[47] Although the project manager fully accepts that reporting is part of his job description, the actual requirements do take time while contributing little added value to the project.

Chapter 4: When Simplicity and Complexity Meet

> *In concentrating on things rather than synthesis, we invite the effects of what has been called "The law of unintended consequences." What will be the effects of highly complex, interrelated systems performing under extreme stress? And can technology allow us to dispense with the Clausewitzian concept of battle, an environment dominated by chaos and friction?*[48]

> *Too much planning can lead to chaos, but so too would too little, and more directly*[49]

What happens when a plan becomes overtaken by events? After the first discontinuity, the cascade of actual chaotic events grow further away from the planned behaviors until the end state is quite different from the intent. A centrally constructed plan with a strict timeline can become "cannon fodder" in a complex, adaptive environment. The main problem with the Y2K Action Plan is that it tries to impose a linear, mechanistic management model on an adaptive, complex environment.

Government bureaucracies, of which the Defense Department is one of the largest, still cling to centrally controlled plans, even when such plans become demonstrably unworkable in a complex environment. One possible reason for this phenomenon is that a centralized plan, through its formal structure, gives the hierarchy the illusion of control in an ambiguous environment. Another reason may be that government planners, especially Army planners, have grown up in a world where every operation, regardless of what it is, requires a plan. In this sense, they plan merely out of habit. Complex systems, however, upset planners' timelines

and intents. Unexpected results, emergent realities, new problems, and "bugs" occur during any adjustment of a complex system, typically leaving the original plan in a shambles.

The Army's original plan for dealing with Y2K followed neat lines of responsibility and an orderly timeline. Unexpected testing requirements and new problems have upset the timeline. Connectivity issues which cross jurisdictional boundaries have clouded the chain of responsibility. The Action Plan, revised in 1998, still delineates actions which needed to occur 1996, two years earlier. This fact alone reduces the utility of the Action Plan for many Army agencies. If the subordinate agency didn't start early enough, then they must modify the plan significantly before they can even use it. Does this mean that planning in a complex environment is futile? Is a detailed plan a waste of the planner's time and effort if everything is going to change anyway? The answer is not to do away with planning, but to change it. An examination of the nature and purpose of planning is in order.

At its core, planning is nothing more than a method of communicating and controlling. Plans inform the organization's members about the intended strategy and its consequences, and to specify what behaviors are expected of particular units and individuals in order to realize the strategy.[50] The plan cannot completely control the outcome of strategy, especially in a complex environment. It can only specify the direction and control some of the actions. A plan will cause some

intended result to occur but, more often than not in a complex system, it will also cause many unintended results to occur, causing planners and operators to revise or create new strategies to deal with the new realities. These are what Mintzberg refers to as "emergent strategies."[51]

Emergent strategies arise from an organization's experience in dealing with unexpected problems and results that were not covered by the original plan. They represent an organization's ability to learn and adapt to a complex environment. When blended with the original, or "intended" strategy, they form the organization's "realized" strategy, or final plan. Plans that rely solely on intended strategy become too inflexible to respond to the fluidity and ambiguity of complex environments. Plans that rely only on emergent strategy, on the other hand, are impossible to control. What is needed, therefore, is a mix of the two. All real-world strategies need to mix these in some way – to attempt to control without stopping the learning process. Effective strategies "mix these characteristics in ways that reflect the conditions at hand, notably the ability to predict as well as the need to react to unexpected events."[52] The best practical means for mixing the two strategies is to have both planners and line managers in on the planning process.

Planners write the broad outlines of the organization's actions and establish boundaries. Line managers are closer to the action and see first-hand the results of the plan and can provide feedback to the planners to let them know if the parameters are unrealistic. The Y2K Action Plan

does have a feedback mechanism of sorts: the Y2K database. This feedback cycle, however, is intended primarily for reporting remediation progress rather than changing the remediation strategy. It is mainly for tracking program status and costs rather than illuminating basic problems with the original plan.

The Action Plan, therefore, does not represent a blend of intended and emergent strategy. It is merely a method of performance control that has, to some extent, become irrelevant in the constantly changing Y2K remediation environment. To become truly effective, the Action Plan would need to incorporate the relevant experiences of each project manager working through the unique problem sets of each individual system. Bringing the line managers into the planning process would also serve to illuminate the complex interconnections of the repair process and aid in solving connectivity issues in a standard manner. Extrapolating these management requirements out to the level of general management of technological innovation in Army systems brings to the fore two critical implications.

The first implication is that planning for changes in complex technological systems requires time; not only for planning, but also for the actual process itself. The case study of the ASAS Y2K remediation, unfortunately, does not offer any answers to the problem of time. Time management in Y2K remediation in general is a unique problem because its completion date is fixed. The only answer might have been to start

earlier. This is an important point. For a complex project requiring major resources, leaders and planners must have the foresight to begin the project early enough to permit the organization to learn while attempting to implement the plan. Line managers must begin work early in the process, even before the higher echelon's plan is fully developed. They must start early because their initial efforts will produce the information needed to fully develop the central plan. This is how an organization learns. If the subordinate agencies wait for a fully developed plan before executing their own work, two things will happen. First, the central plan will never receive the information it needs to become fully developed and will therefore be incomplete, vague, and of little relevance to the project at hand. Second, if the line managers wait, they will begin too late to bring the plan, with all of its emergent, unexpected phenomena, to fruition.

Expecting to inventory, assess, repair and modify every automated system in the Army for the century change in less than three years is obviously unrealistic. If it is obvious that the line managers cannot implement the plan in the time allotted, then the planners must change the plan to reflect the new reality. More emphasis on triage and contingency plans would be appropriate in this case.

There are other time-related considerations when blending deliberate planning with emergent events in a complex environment. Bringing planners and line managers together, wrestling with the complexities of the project, working out and perhaps negotiating clear

lines of responsibility and coming up with a clear plan based on the needs of the organization as well as the experience of the operators all take a significant amount of time. Furthermore, the line managers then will require sufficient time to build their complex systems properly. Starting from the simple and building to the complex with adequate testing at each level takes time, especially as new, unexpected events occur which add to the length of the process. Finally, the plan must allow for additional time to incorporate modifications to the plan based on the results of these unexpected events. This is closely related to the second implication: control.

Central planners in a hierarchical organization, like the Army, have been accustomed to having the ability to control the actions of the organization. The Army is especially wedded to the concept of control because of its commitment to a well-defined chain of command and clearly delineated responsibilities at every level. Complex systems, however, resist centralized control. Adaptive environments change and evade control when the parameters set on them are not adequate or relevant. Complexity produces unpredictability. Hierarchical managers combat this by overlaying more controls on the system. They add more layers of complexity to control unforeseen behaviors; producing more unpredictability in the process, or producing rigidity.[53] This is problematic. According to Kelly, "any centrally controlled complexity is unstable and inflexible."[54] If there is an answer to this dilemma, it lies in training

planners and upper level executives. When planners and executives come to realize that emergent strategies will mix with and change the original plan, then they will know that they cannot hope to control every detail of the process. They can lay out the guidelines but the details will remain the responsibility of lower level managers and operators. The end result of the action may look somewhat different than the planners' original intent, but as long as the end result is mission accomplishment, line managers and planners have met the intent of the plan.

Conclusion

> *All modern systems and certainly program management have become more complex, not merely in scale but in kind. Non-linear feedback systems are the norm, yet our education system has provided mainly static methodology for considering these dynamic systems.*[55]

It is difficult for contemporary executives and planners to come to grips with complexity. Raised in a simpler, mechanistic world, these individuals are uncomfortable with environments they cannot reliably predict or control. The intricate, interrelated nature of complex systems defies analysis. Trying to break a complex system into neat, well-defined components ignores the main characteristic of complex systems: the relationships between the components. Still, planners will attempt to ignore complexity and try to fit complex problems into simple boxes.

There are a number of impediments to systems-thinking, or the ability to understand and manage complexity. First, the relatively short period of time that planners spend in their jobs gives rise to a fast-paced frame of reference in which the higher-order consequences of actions don't emerge until after the planner has moved on. Second, the planners' hectic schedule causes them to "live in the fray" and react to the fire at hand. Third, planners' tend to frame problems in terms of visible objects rather than relationships. Finally, because successful planners, executives, and systems have all reached their positions through an evolutionary survival process, they tend to value action over deliberation and reflection.[56] Dealing with complexity is not an easy task. As Rolf

Clark, an educator in systems management relates,

> ...interrelationships of modern systems are too complex for the human mind. They are too complex for the sophisticated analytical techniques taught in academic institutions as well...We need to stop trying to "solve" and instead learn to explore.[57]

Taking the time to explore and reflect on the nature of complex systems does not come naturally to executives and planners, especially officers raised in the action-oriented world of military operations. Recognizing that not all systems respond to direct control is the first step into the greater world of complexity.

Executives and planners need to become familiar with the characteristics of complexity. Plans that define success or failure based on simple performance criteria will miss the learning that takes place in an organization during the execution of a complex task. By considering the effects of complexity on planning, organizations can benefit from the emergent lessons of complex actions in complex systems. By "clutching the reigns" a little less tightly and allowing for additional time for the organization to learn from and adapt to emergent realities, high level executives and planners can continue to guide the Army into the complex world of the next millennium.

[1] Jim Lord, *U.S. Government is botching Y2K repairs*, (www.garynorth.com/y2k/detail_.cfm/ 2645), 22 Sep 98.

[2] Ed Yardeni, "Who gave us the Two Digit Year?," *Dallas Morning News*, (www.dallasnews.com /technology-nf/techbiz1.htm,) 4 Oct 98

[3] Ed Yourdon, *Timebomb 2000*, (www.yourdon.com,) Chapter 2, p. 1

[4] Harry S. White, Jr., "Who gave us the Two Digit Year? The Pentagon," *Dallas Morning News*, (www.dallasnews.com/ technology-nf/techbiz1.htm,) 4 Oct 98

[5] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (www.imabbs. army.mil/ army-y2k/APRev2/AP_Sect1-7.htm, June 1998,) p. 2

[6] United States Army, *Weapon Systems*; (Washington, DC: U.S. Government Printing Office, 1998,) p. 13

[7] United States Army, FM 34-25-3, *All Source Analysis System and the Analysis Control Element*; (HQ, Department of the Army: 3 Oct 95,) p. 1-2

[8] Theodore G. Chopin, "ASAS System Update and Planned RWS Field and Institutional Support," *Military Intelligence Professional Bulletin*, (Vol 22, No. 4, Oct-Dec 96,) pp. 10-12

[9] Letter from PMO ASAS (Mr. John Muccio), 30 Sep 1998

[10] Kevin Kelly, *Out of Control: The new biology of machines, social systems, and the economic world*, (New York: Addison Wesley Publishing Company, 1994,) p.198

[11] Ed Yourdon, *Timebomb 2000*, (www.yourdon.com,) p. 2

[12] Margaret Wheatley and Myron Kellner-Roges, "Turning to One Another: The Possibilities of Y2K," *Y2K Citizen's Action Guide*, (www.utne,com/y2k/turning.html, 1 Nov 98), Chapter 9, p. 1

[13] PMO ASAS, *Y2K Database (Y2KDB) Evaluation Memorandum*, (www.army.mil/armyy2k/ SysFails.CFM?, 11 Nov 98)

[14] Henry Mintzberg, *The Rise and Fall of Strategic Planning*, (New York: the Free Press, 1994,) p. 348

[15] Ibid., p. 106

[16] Ibid., p. 13

[17] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) section 1

[18] Henry Mintzberg, *The Rise and Fall of Strategic Planning*, (New York: the Free Press, 1994,) p. 78

[19] Ibid., p. 78

[20] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) section 3

[21] Ibid., section 8

[22] Henry Mintzberg, *The Rise and Fall of Strategic Planning*, (New York: the Free Press, 1994,) pp. 223-224

[23] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) section 9

[24] Henry Mintzberg, *The Rise and Fall of Strategic Planning*, (New York: the Free Press, 1994,) p. 260

[25] Ibid., p. 264

[26]DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) section 14

[27]PMO ASAS, *Y2K Database (Y2KDB) Evaluation Memorandum*, (www.army.mil/armyy2k/ SysFails.CFM?, 11 Nov 98)

[28] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) Appendix B

[29] Government of California, *Y2K White Paper*, (www.year2000.ca.gov/correspondence/ CA2000WhitePaper.asp, 16 October 1996,) pp. 16-18

[30] Steve Tomczak, *Nothing Happens Until the Fat Lady Sings*, (http//greenspun.com/bboard/q-and-a-fetch-msg.tcl?msg_id=000E7H, 3

Nov 98,) p. 1

[31] Ibid., p. 1

[32] Ibid., p. 1

[33] Corrine Gregory, "Efficient Testing Approaches," *Millennium Journal*, (www.data-dimensions.com/html/milj54.htm, 16 Apr 98,) p. 1

[34] Tom Puller-Strecker, "Banks Aim to Give Y2K assurances," *Infotech Weekly*, (www.infotech. co.nz/november_30/nxib.html, 1 Dec 98,) p. 1

[35] Letter from PMO ASAS (Mr. John Muccio), 30 Sep 98

[36] Christopher Price and Avi Machlis, "Flaws found in Y2K Conversions," *San Jose Mercury News*, (www5.mercurycenter.com/business/tech/ docs/081336.htm, 19 Nov 98,) p. 1

[37] Ibid., p. 2

[38] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) section 14

[39] Ibid., section 11

[40] Christopher Price and Avi Machlis, "Flaws found in Y2K Conversions," *San Jose Mercury News*, (www5.mercurycenter.com/business/tech/ docs/081336.htm, 19 Nov 98,) p. 3

[41] Department of Defense, Office of the Inspector General, "Year 2000 Certification of Mission-Critical DoD Information Technology Systems", *Report No. 98-147 (PFD)*, (5 Jun 98,) p. 1

[42] Heinz Pagels, *The Dreams of Reason: the Computer and the Rise of the Sciences of Complexity*, (New York, Toronto: Bantam Books, 1984,) p. 67

[43] Kevin Kelly, *Out of Control: The new biology of machines, social systems, and the economic world*, (New York: Addison Wesley Publishing Company, 1994,) p. 41

[44] Phone conversation with Troy Harriet, CECOM ASAS project manager, 3 Dec 98

[45] DISC4, *The U.S. Army Year 2000 (Y2K) Action Plan, Revision II*, (June 1998, www.imabbs. army.mil/army-y2k/APRev2/AP_Sect1-7.htm,) section 6

[46] Phone conversation with Troy Harriet, CECOM ASAS project manager, 3 Dec 98

[47] Ibid.

[48] McKenzie, Kenneth, "Beyond Luddites and Magicians, Examining the MTR," *Parameters*, (Vol 25, No. 2, Summer 95,) p. 18

[49] Henry Mintzberg, *The Rise and Fall of Strategic Planning*, (New York: the Free Press, 1994,) p. 416

[50] Ibid., pp. 351-353.

[51] Ibid., p. 25

[52] Ibid., p; 25

[53] Chris C. Demchak, *Military Organizations, Complex Machines: Modernization in the U.S. Armed Services*, (Ithaca, NY: Cornell University Press, 1991,) p. 4

[54] Kevin Kelly, *Out of Control: The new biology of machines, social systems, and the economic world*, (New York: Addison Wesley Publishing Company, 1994,) p. 42

[55] Rolf Clark, "Education Gaps in a Complex World," *Program Manager*, (Vol 20, No. 4, Jul/Aug 91,) p. 30

[56] Ibid., p. 31

[57] Ibid., p. 31

BIBLIOGRAPHY

Beaumont, Roger A., *War, Chaos, and History*; (Westport, Conn: Praeger, 1994)

Chacko, George K., *Technology Management: Applications to Corporate Markets and Military Missions*; (New York: Prager, 1988)

Churchman, C., *The Systems Approach and its Enemies*; (New York: Basic Books, 1979)

Demchak, Chris C., *Military Organizations, Complex Machines: Modernization in the U.S. Armed Services*; (Ithaca, NY: Cornell University Press, 1991)

Doerner, Dietrich, *The Logic of Failure*; (New York: Metropolitan Books, 1991)

Kelly, Kevin, *Out of Control: The New Biology of Machines, Social Systems, and the Economic World*; (New York: Addison Wesley Publishing Company, 1994)

Mintzberg, Henry, *The Rise and Fall of Strategic Planning*; (New York: The Free Press, 1994)

O'Toole, James, *The Executive's Compass: Business and the Good Society*; (New York: Oxford University Press, 1993)

Pagels, Heinz, *The Dreams of Reason: The Computer and the Rise of the Sciences of Complexity*; (New York, Toronto: Bantam Books, 1984)

Piattelli-Palmarini, Massimo, *Inevitable Illusions*; (New York: Wiley and Sons, 1994)

Senge, Peter M., *The Fifth Discipline*; (New York: Doubleday, 1994)

Simon, Herbert A., *The Sciences of the Artificial*; (Cambridge, MA: MIT Press, 1996)

Waldrop, M. Mitchell, *Complexity*; (New York: Simon and Schuster, 1990)

Whittington, Richard, *What is Strategy and What Does it Matter?;* (London, New York: Routledge, 1993)

ARTICLES:

Chopin, Theodore, "ASAS: New Ways to Leverage Human Analytical Power," in *Military Intelligence Professional Bulletin*; vol 22, no 3 (Jul-Sep 96) pp. 10-12

Chopin, Theodore, "ASAS System Update and Planned RWS Field and Institutional Support," in *Military Intelligence Professional Bulletin*; vol 22, no 4 (Oct-Dec 96) pp. 11-13

Clark, Rolf, "Education Gaps in a Complex World," in *Program Manager*; vol 20, no 4 (Jul/Aug 91) pp. 28-31

Fallon, Michael, "ASAS in Operation: Joint Warfighter Interoperability Demonstration," in *Military Intelligence Professional Bulletin*; vol 22, no 3 (Jul-Sep 96) pp. 13-15

Krepinevich, Andrew, "Keeping Pace with the Military-Technological Revolution," in *Issues in Science and Technology*; vol 10, no 4 (Summer 94) pp. 23-29

McKenzie, Kenneth, "Beyond Luddites and Magicians: Examining the MTR," in *Parameters*; vol 25, no 2 (Summer 95) pp 17-19

Schneider, James J., "Black Lights: Chaos, Complexity, and the Promise of Information Warfare," in *Joint Force Quarterly*; no 15 (Spring 97) pp. 21-28

MILITARY PUBLICATIONS:

Department of Defense, *Year 2000 Management Plan, Version 2.0*; (Wash DC: ASD for C3I, www.dtic.mil/c3i/y2k, Apr 94)

United States Army, FM 34-25-3, *All Source Analysis System and the Analysis Control Element*; (HQ, Department of the Army: 3 October 1995)

United States Army, *Weapon Systems*; (Wash DC: US Government Printing Office, 1998)

United States Army DISC4, *Year 2000 Action Plan, Revision II*; (Wash DC: ADCS-CIO, www.imabbs.army.mil/army-y2k, June 98)

INTERNET RESOURCES AND ARTICLES:

Campbell, William H., LTG, "Year 2000 Progress and Issues," *DISC4 Webpage*; (imabbs.army.mil/y2k-army/junepolicy.htm, 8 Jun 98)

Government of California, *Y2K White Paper*, (www.year2000.ca.gov/correspondence/CA2000WhitePaper.asp, 16 Oct 96)

Gregory, Corrine, "Efficient Testing Approaches," in *Millennium Journal*; (www.data-dimensions.com/html/milj54.htm, 1 Dec 98)

Lord, Jim, *U.S. Government is Botching Y2K Repairs*; (www.garynorth.com/y2k/detail_.cfm/2645, 22 Sep 98)

Price, Christopher and Machlis, Avi, "Flaws Found in Y2K Conversions," in *San Jose Mercury News*; (www5.mercurycenter.com/business/tech/docs/081336.htm, 19 Nov 98)

PMO ASAS, *Y2K Database (Y2KDB) Evaluation Memorandum*; (www.army.mil/armyy2k/SysFails.CFM?, 11 Nov 98)

Tomczak, Steve, *Nothing Happens Until the Fat Lady Sings*; (http//greenspun.com/bboard/q-and-a-fetch-msg.tcl?msg_id=000E7H, 3 Nov 98)

Wheatley, Margaret and Kellner-Rogers, Myron, "Turning to One Another: The Possibilities of Y2K," in *Y2K Citizen's Action Guide*; (www.utne.com/y2k/turning. html, 1 Nov 98)

White, Harry S., Jr., "Who Gave Us the Two Digit Year? The Pentagon," in *Dallas Morning News*; (www.dallasnews.com/technology-nf/techbiz1.htm, 4 Oct 98)

Yourdon, Ed, *Timebomb 2000*; (www.yourdon.com, Dec 97)

OTHER SOURCES:

Letter from PMO ASAS (Mr. John Musio) 30 Sep 98

Phone Conversation with CECOM ASAS Project Manager (Mr. Troy Harriet) 3 Dec 98